WVS•VORTEX(1)•620-DIFF/DOC

DIFFERENCES BETWEEN VARIAN'S 620 EMULATOR AND W. V. SNYDER'S 620 EMULATOR.

1. ILLEGAL INSTRUCTION CONTENTS CAUSE ERRATIC RESULTS IN THE VARIAN 620 EMULATOR. IN MY EMULATOR, ILLEGAL INSTRUCTIONS CAUSE AN INTERRUPT TO A PARAMETRICALLY SPECIFIED ADDRESS, NOW SET TO 2.
2. THE ADDRESSES USED BY FIRMWARE FOR RE-ENTRY ARE DIFFERENT.
3. IN MY EMULATOR, IMMEDIATE INSTRUCTIONS CAN BE INDEXED AND INDIRECTLY ADDRESSED SIMILARLY TO THE METHOD USED FOR EXTENDED INSTRUCTIONS. BITS 7, 1 AND 0 ARE USED FOR THIS ADDRESSING AS FOR EXTENDED ADDRESSING: IF BITS 1 AND 0 ARE 00, 01 OR 10, THE OPERAND IS INDEXED WITH THE P, X OR B REGISTER, AND INDIRECT ADDRESSING IS PERFORMED IF SPECIFIED. IF INDIRECT ADDRESSING IS SPECIFIED, BIT 7 IS USED TO CONTROL PRE- OR POST-INDEXING. THE FINAL RESULT (WHICH WOULD BE AN ADDRESS FOR AN EXTENDED INSTRUCTION) IS THEN THE OPERAND. THUS, NEGATIVE OPERANDS ARE NOT POSSIBLE. IF BITS 1 AND 0 ARE 11, THE INSTRUCTION OPERATES EXACTLY AS THE VARIAN 620 INSTRUCTION. NOTE THAT "DASMR" ASSEMBLES BITS 1 AND 0 AS 00. MY EMULATOR MAY BE MADE TO EXECUTE INSTRUCTIONS SO ASSEMBLED BY EXCHANGING TWO PAIRS OF INSTRUCTIONS.
4. EXTENDED INSTRUCTIONS WITH THE "M" FIELD EQUAL 0 OR 8 ARE ILLEGAL INSTRUCTIONS IN THE VARIAN EMULATOR. IN MY EMULATOR, THEY ARE DOUBLE LOAD AND DOUBLE STORE, RESPECTIVELY.
5. SHIFT INSTRUCTIONS HAVING BOTH THE "LONG" AND "A" BITS SET ARE ILLEGAL INSTRUCTIONS IN THE VARIAN EMULATOR. IN MY EMULATOR, THEY ARE TREATED EXACTLY THE SAME AS THEIR COUNTERPARTS WITH ONLY THE "LONG" BIT SET.
6. THE INDEXED JUMP ("IJMP") INSTRUCTION CAN ONLY BE POST-INDEXED IN THE VARIAN EMULATOR. IN MY EMULATOR, BIT 2 CAN BE USED TO CONTROL WHETHER PRE- OR POST-INDEXING IS PERFORMED, AS CAN BE DONE WITH EXTENDED INSTRUCTIONS. NOTE THAT "DASMR" ASSEMBLES "IJMP" WITH BIT 2 EQUAL TO ZERO, AND IF SOMEBODY HAS FOUND A WAY TO USE A POST-INDEXED INDIRECT "IJMP", IT WILL NOT EXECUTE PROPERLY IN MY EMULATOR. MY EMULATOR CAN BE MADE TO EXECUTE INSTRUCTIONS SO ASSEMBLED, IN THE SAME WAY AS VARIAN INTENDED BY EXCHANGING TWO PAIRS OF INSTRUCTIONS. IN ADDITION, IN MY EMULATOR, BITS 4 AND 3 CAN BE USED TO SELECT TRANSFER OF THE CURRENT P REGISTER CONTENTS TO THE B OR X REGISTER ("IJSR"), OR TO PERFORM EXECUTE REMOTE ("IXEC"). THE "JSR" INSTRUCTION COULD BE DELETED.
7. IN THE VARIAN EMULATOR, EXACTLY ONE OF THE REGISTER SELECTION BITS MUST BE SET FOR THE SKIP IF REGISTER EQUAL INSTRUCTION ("SRE") TO WORK PROPERLY. IN MY EMULATOR, THE REGISTER SELECTION BITS WORK EXACTLY AS THEY DO IN THE REGISTER-TO-REGISTER TRANSFER INSTRUCTIONS.
8. TO FACILITATE THE USE OF COMMON I/O ROUTINES FOR IDENTICAL DEVICES HAVING DIFFERENT DEVICE ADDRESSES, MY EMULATOR INCLUSIVE-OR'S THE LOW-ORDER 6 BITS OF REGISTER 3 (WHICH IS ADDRESSED PARAMETRICALLY) WITH ALL DEVICE ADDRESSES SENT TO THE I/O BUS. AN INSTRUCTION HAS BEEN PROVIDED TO LOAD THIS REGISTER. SINCE THE HIGH-ORDER 10 BITS MUST BE ZERO, THE HALT LOOP WILL NOT LOAD THESE BITS.
9. MY EMULATOR DOES NOT REQUIRE R3=ZERO OR R5=ONES.
10. IN VARIANS EMULATOR, THE ADDRESS COMPUTATION FOR EXTENDED INSTRUCTIONS IS 33 MICRO INSTRUCTIONS. IN MY EMULATOR IT IS 11. THERE ARE 3 SLIGHTLY DIFFERENT VERSIONS OF THIS ROUTINE IN MY EMULATOR (EXTENDED, IMMEDIATE, IJMP). THE COMPUTATION IS ONE MICRO FASTER PER POST-INDEXED INDIRECT ADDRESS LEVEL AFTER THE FIRST THAN IS VARIANS. AN ERROR IN THE DESIGN OF THE PROCESSOR WAS DISCOVERED DURING CHECKOUT OF THE EXTENDED ADDRESS COMPUTATION: IF THE MEMORY IS ACTIVE (PREVIOUS MEMORY OPERATION SYNCHRONIZED BY STARTING A NEW MEMORY OPERATION), AND A CONDITIONAL MEMORY OPERATION WHICH DEPENDS ON MIL15 IS INITIATED SUCCESSFULLY, A BRANCH DEPENDING ON MIL15 WILL NOT WORK PROPERLY. THE REASON IS THAT THE MEMORY INPUT LATCH IS NOT A TRUE TYPE D LATCH. THE OUTPUT FOLLOWS THE INPUT WHILE

```
     57          THE CLOCK IS HIGH, AND LATCHES ON THE CLOCK TRANSITION.  A TRUE TYPE D LATCH
     58          WOULD HAVE A MASTER-SLAVE ARRANGEMENT SO THAT THE OUTPUT WAS INDEPENDENT OF
     59          THE INPUT UNTIL THE CLOCK TRANSITION.
     60      11. MULTIPLY AND DIVIDE INSTRUCTIONS ARE FASTER IN MY EMULATOR UNDER CERTAIN
     61          CIRCUMSTANCES.
     62      12. THERE APPARENTLY IS A POSSIBILITY THAT THE DMA CAN GENERATE A SIGNAL WHICH
     63          LOOKS LIKE AN I/O INTERRUPT.  WHEN THE MICROPROGRAM GETS TO LOCATION X'D7
     64          (THE SAME AS IN VARIAN'S EMULATOR) AND STARTS THE I/O MICROPROGRAM TO
     65          SERVICE THE INTERRUPT, THE NEXT MICRO IS FETCHED USING NORMAL ADDRESSING
     66          INSTEAD OF INTERRUPT ADDRESSING (WHICH WOULD TRANSFER TO X'D1).  IF THIS
     67          SITUATION OCCURS, MY EMULATOR WILL DEAL WITH IT USING ONE LESS MEMORY
     68          REFERENCE THAN VARIAN'S EMULATOR.
     69      13. MY EMULATOR MICROPROGRAM IS EXTENSIVELY COMMENTED AND RATIONALLY
     70          ORGANIZED, AND HENCE IS FAR MORE READABLE THAN VARIAN'S EMULATOR, WHICH
     71          HAS ONLY 2 COMMENTS AFTER THE FIRST EXECUTABLE STATEMENT.  ALSO, MANY OF
     72          THE QUANTITIES USED IN THE PROGRAM, SUCH AS INTERRUPT MASKS AND REGISTER
     73          ASSIGNMENTS ARE PARAMETRICALLY SPECIFIED.

CPU:.022    CTP:.004    SUPS:.940

@HDG,N

@BRKPT PRINT$
```